REMARKS

Claims 1 through 20 are presented in the instant patent application. Claims 1, 8, and 15 are the independent claims. Claims 2 through 7 depend from Claim 1. Claims 9 through 14 depend from independent Claim 8. Claims 16 through 20 depend from Claim 15.

1 Rejections under 35 U.S.C. § 112, second paragraph

In the Action at page 3, Claim 20 is rejected under 35 U.S.C. § 112, second paragraph as failing to particularly point out and distinctly claim the subject matter which the Applicants regard as the invention. Specifically, Claim 20 is rejected as there is insufficient antecedent basis for the limitation "given software application domain terminology". Applicants believe this is now moot in view of the amendment to the base claim from which Claim 20 depends. Base Claim 15 now recites this element providing proper antecedent basis, and Claim 20 depends from Claim 15, and thus, now Claim 20 is clear, and definite, and conforms with 35 U.S.C. § 112, second paragraph. Reconsideration and withdrawal of the rejection are respectfully requested.

2. Rejections under 35 U.S.C. § 101

In the Action, Claims 1 through 7 are rejected under 35 U.S.C. § 101. The claims are rejected since the invention is supposedly directed to a computer apparatus of functional descriptive material, *per se*, and is therefore alleged to be non-statutory.

Specifically, the Office contends that such claimed computer programs do not define any structural and functional interrelationships between the programs and other elements of the computer, which permit the functionality to be realized. Applicants must respectfully disagree. Claim 1 recites, *inter alia,* a data server. Clearly, this is a computer element which defines a structural and functional interrelationship between a program and the rest of a computer.

Claim 1 recites computer components that define structural and functional interrelationships between programming elements, and this permit the functionality to be realized. Also included are an editor (defining class views and a composite class view), and a multi-tiered data model.

The present invention uses these components to provide ways to provide enhanced system control over process and plant engineering routines. The data server, editor, and model enable the sharing of data through interfaces. This permits the operation of the multi-tier data model, and permits the operation of the subject program or software applications.

The present disclosure provides precise control of a physical plant, while keeping class views in a given software application domain terminology, and also while keeping these sets of programs separate in a multi-tiered manner for modeling.

Thus, not only does the claimed invention provide a concrete and tangible result, it provides an advantage over other ways of modeling behavior. This additional tangible result is not disclosed in the cited art. For the foregoing reasons, Claims 1-7 as now amended are believed to be directed to statutory subject matter. Thus, reconsideration and withdrawal of the 35 U.S.C.§ 101 rejection of Claims 1-7 are respectfully requested.

In an abundance of clarity and to expedite prosecution, Claim 1 is now amended to further emphasize the structural and functional interrelationships between computer program parts and the rest of the computer. With this amendment, Claim 1 is believed to recite statutory subject matter in the manner consistent with Pg. 4 of the Office Action at hand. Acceptance is respectfully requested.

3. Rejections under 35 U.S.C. § 102(e)

Claims 1-6, 8-13, and 15-19 are rejected in the Action under 35 U.S.C. § 102(e) as being anticipated by United States Patent No. 7,047,518 B2 to Little *et al,*. (hereinafter "Little"). In response, Applicants respectfully traverse the rejection as Little does not disclose or suggest all of the elements of the independent claims.

Little discloses a system for software application development and modeling. Little discloses a system that automatically generates software code in accordance with some preset wishes of the software developer. See Little at Column 1, lines 53 through 63.

Little also discloses using a Unified Modeling Language, or a non-proprietary language, which provides software architects with a standardized language for specifying, constructing, visualizing, and documenting the artifacts of a complex software system. See Little at Column 2, lines 10 through 23. The modeling development system uses Unified Modeling Language (hereinafter "UML") object modeling. See Column 11, lines 23 through 45. The UML is a standard language for specifying, visualizing, constructing, and documenting the artifacts of software systems, as well as for business modeling and other non-software systems. UML provides graphical notations to express the design of software projects. Little discloses creating a data entity framework at step 302 of FIG. 14. The data entity groups are created using an Expert System menu option (step 304). These data entity groups create a set of relational tables and relational view classes, which represent RDBMS tables, and views and customized definitions. See Column 16, line 55 through Column 17, line 13.

Next at step 306, customized access classes are created in FIG. 14. Here, the programmer creates relational table classes and a class diagram to show the relationship between classes. Little does disclose that a composite view class is created at step 308. Dependent relationships values are established between multiple tables. See Column 17, line 25 through line 36.

For every relational table, view, or composite class in Little, a data entity package is created in the program, and maps to the relational table and the relational view class (Step 312). See Column 18, lines 6 through 26. **Little also discloses that the system generates C++ or Java classes that implement the model they have created.** See Column 19, line 27 through 31.

Finally, at Column 19, lines 16 through 31, Little discloses that a code generation is conducted, which prepares the model for implementation. This creates data groups for a component view, and all details, such as, physical source files, make files, and libraries, are populated in all corresponding model components.

Further, a number of directories for this code is created. **Generation of data access C++ source files, are then made.** See Column 19, line 27 through 31.

Little does not disclose or suggest a computer apparatus that includes a consolidation of the Class Views resulting in the creation of the composite class view, which is an amalgamation and rationalization of the individual class views, and the class views remain in the application domain terminology, as presently amended in base Claims 1, 8 and 15. Support for this amendment can be found in the patent application as originally found in various location including, for example, at page 8, lines 1 through 4 of the patent application as originally filed.

Little uses UML or a general purpose modeling language that includes a graphical notation that is used to create not a multi-tiered model, or any 'consolidated' composite class view, but instead an abstract model of the system. This would result in a change of formats of the original data, which is exactly what the present application seeks to avoid. Clearly this is not a class view that remains in the application domain terminology as claimed. See Little at step 308, FIG. 14, and at Column 17, lines 25 through 36 showing multiple table operation at lines 23 through 36, or creation of an "enhanced design UML model". See Little at Column 30, lines 66 through 67.

This is especially advantageous since the combined system (data server and class editor) enables the sharing of original application data with process plant engineering routines and programs in the application domain terminology. Claims 2 through 6 are also patentable as these claims depend from Claim 1. Independent Claims 8, and 15 are patentable for reasons similar to those argued above for Claim 1. Claims 9 through 13 are also patentable as these claims depend from independent Claim 8. Claims 16-19 are also patentable as these claims depend from independent Claim 15 Accordingly, the 35 U.S.C. § 102 rejection of Claims 1 through 6, and 8 through 13, and 15-19 is overcome. Reconsideration and withdrawal of the rejection are respectfully requested.

4. Rejections under 35 U.S.C. § 103(a)

In the Action, Claims 7, 14, and 20 are rejected under 35 U.S.C. § 103(a) as being unpatentable as obvious over Little. Applicants traverse the rejection, and state that one of ordinary skill in the art would not be motivated to hypothetically modify the primary reference as indicated by the Office at page 15-18 of the Action.

Applicants contend that one of ordinary skill in the art having Little hypothetically would not have the class views remaining in the application domain terminology, as presently amended in base Claims 1, 8 and 15. Instead, applicants rebut the contentions of the Office and state that after reviewing Little the UML would **generate data access C++ source files.** See Column 19, line 27 through 31.

Applicants submit that these teach away from one another, which is a strong presumption in favor of patentability of Claims 7, 14 and 20, which depend from respective base Claims 1, 8 and 15. The 35 U.S.C § 103(a) rejections of Claims 7, 14, and 20 are believed to be overcome and should be withdrawn. Reconsideration and withdrawal of the rejection are earnestly solicited.

CONCLUSION

In view of the above amendments and remarks, it is believed that all claims are in condition for allowance, and it is respectfully requested that the application be passed to issue. If the Examiner feels that a telephone conference would expedite prosecution of this case, the Examiner is invited to call the undersigned.

Respectfully submitted,

HAMILTON, BROOK, SMITH & REYNOLDS, P.C.

Mary Lou Wakimura by
By , Esq #46,605
Mary Lou Wakimura
Registration No. 31,804
Telephone: (978) 341-0036
Facsimile: (978) 341-0136

Concord, MA 01742-9133
Date: 9/24/2007